

22/10/2016

العنوان

CSIS

محاضرة 6

## \* Transport Layer

Its output are segments that are sent to network layer

- Transport layer is on end systems

# Logical process (طبقات منطقية)   
 Applications → Logical Comm. Layer → Transport layer → Net. Layer → End Systems

(app. processes → app. processes → ... → app. processes) اoward application processes

مع مساعدة OS → على كل end systems يجري Logical Comm. Layer و Transport layer و Net. Layer و End Systems

\* Network Layer → datagram output

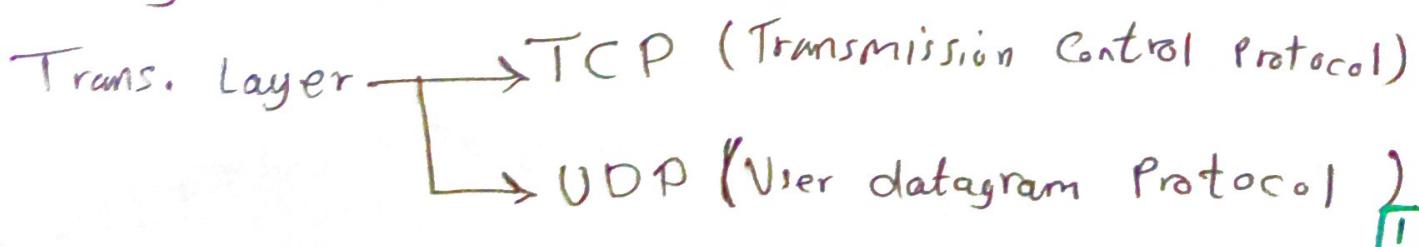
\* Transport Layer → segment output

بروتوكول إنترنت (internet protocol) IP هو بروتوكول الشبكة Net. Layer يخاطب المكونات المختلفة و ينقلها للاتصالات Layer

\* Network Layer protocol is unreliable ("best-effort")  
does not check Bandwidth, congestion, packet Loss

\* Complexity is in Transport Layer instead of Net. layer

\* it's up to app. developer to choose trade off between security or speed.



\* App developer choose between UDP and TCP

- ① ~~Fast~~ (High throughput, low delay, no security)  
② ~~Slow~~

Check 3.4 for comparison UDP vs TCP

TCP is Slower Than UDP because of

- ① Congestion control  
② Flow control  
③ Error checking

TCP (reliable but slow)  
UDP (unreliable but fast)

we can't be reliable in Net Layer IP

Transport Layer is unreliable & reliable  
(Complexity) need more reliability

Algorithms بحسب وظائف لـ Transport Layer

\* Transport Layer does Multiplexing/demultiplexing  
for its segments

ما يفعله Segmenting and Joining App message: Mux

وامثلة Net Layer

ـ تابع راسخ

ـ Net Layer segments: جعلها متحدة وتحقيقها  
ـ demux: وفرزها على المستخدمين

Mux → collect and send

demux → receive and ~~distribute~~

- \* we have sockets between App and trans. Layer each socket has ID and is application specific
- \* When received e.g. email; segment has the socket ID so Trans. Layer knows which app to receive the segment.

جعن buffer ي؛ هو Socket ID ~~Segments~~ \*  
أيضاً هو App ID ~~Segments~~ \*  
Segments و Application ID من socket ID \*

~~Segments~~ و Application ID من socket ID \*

Check 3.5 for examples

- \* Trans layer can do both Mux and demux at the same time.

UDP و TCP ~~segment~~ #  
headers معلومات داخل  #  
SRC Port No.  
dest Port No.

32 bit dest. و Source IP 16 طول Port No. \*  
عند developer أو Application ندري سترم لازم يعرف  
ال Port number المطلوب

ال Net Layer فقط ومن ثم يجيء ال IP

headers تعرف بجزء من ال layers \*  
فقط من كلام ابراهيم ، كل رسالة يجيء ال IP من خارج  
ال Net Layer .

ال Client هو Port الى يسْتَعْلِمُ عَلَيْهِ اَو SRC Port على application

Telnet uses UDP , HTTP uses TCP

check 3.7 for example

UDP uses 8 byte header

TCP : 20 byte header



\* UDP is connection-Less , unreliable data transfer  
, no congestion and flow control

\* it only does checksum ( data check ) 16 bit  
by first complement of data.

- يقسم ار 16 بت data

- يجمعها وليجيب ار المجموع First complement

- ونأخذ ار checksum header ليخزن في ار

- ار 16 بت اخافته بار checksum هي التي بينا كل بعدها متساوية ار

- لـ 16 بت اخافته بار checksum او Segments كل انتاج بـ 16 byte  
صيغة الـ 16 بت سوية كلها متساوية وتفهم .

# Length → to know the segment's size

segments can vary in size

اـ 5 - وـ 2 من مكانه في اـ 2 لـ 16 بت تغيير في عدوين ( Checksum من هذين